Shorの素因数分解アルゴリズムにおける計算量の精密な評価 (応用函数解析としての情報数理の研究)

AUTHOR(S):

栗山, 憲; 佐野, 慎太郎; 古市, 茂

CITATION:

栗山, 憲 ...[et al]. Shorの素因数分解アルゴリズムにおける計算量の精密な評価 (応用函数解析としての情報数理の研究). 数理解析研究所講究録 2005, 1452: 206-214

ISSUE DATE:

2005-10

URL:

http://hdl.handle.net/2433/47771

数理解析研究所講究録 1452 巻 2005 年 206-214

Shorの素因数分解アルゴリズムにおける計算量の精密な評価

栗山憲
(Ken Kuriyama)
山口大・工
(Department of Applied Science, Yamaguchi University)

佐野慎太郎
(Shintaro Sano)
山口大・理工
(Graduate School of Science and Engineering, Yamaguchi University)

古市茂
(Shigeru Furuichi)
山口東京理科大・基礎工
(Department of Electronics and Computer Science, Tokyo University of Science in Yamaguchi)

1 はじめに

1994 年に Shor は量子コンピュータを用いた効率的な素因数分解アルゴリズムを発表した [1, 2]. 現在のコンピュータでの大きな整数の素因数分解には膨大な計算量を要することはよく知られており，インターネット等で広く使われている暗号の安全性はこの計算量の大きさに依存している．そこで，量子コンピュータが実現されるとこの種の暗号の安全性が崩壊するとして Shor の研究は注目を浴びた．

Shor のアルゴリズムの計算量は，素因数分解したい数 n を 2 進数表示したときの桁数 $\log_2 n$ についての多項式時間となる．本研究の目的は，この計算量をより精密に評価することである．従来の評価では，素因数分解したい数 $n = {p_1}^{e_1}{p_2}^{e_2}\ldots{p_k}^{e_k}$ に対して，十分大きい確率で正しく素因数分解するために必要なアルゴリズムの実行回数 N は

$$\forall \varepsilon > 0, \quad N \geq \frac{\log(1/\varepsilon)}{\alpha\beta(1-1/2^{k-1})}(\log_2 n)^2$$

であった．これに対して，本研究の精密な評価では $p_i - 1 = 2^{\tau_i}\sigma_i$ $(i = 1, 2, \ldots, k,\ \tau_i \geq 1,\ \sigma_i$ は奇数$)$, $\tau' = \min(\tau_1, \ldots, \tau_k)$, $\tilde{\tau} = \sum_{i=1}^{k} \tau_i$ とすると

$$\forall \varepsilon > 0, \quad N \geq \frac{\log(1/\varepsilon)}{\alpha\beta\left(1 - \frac{1}{2^k-1}\frac{2^k-2+2^{k\tau'}}{2^{\tilde{\tau}}}\right)}(\log_2 n)^2$$

と表すことができる．これにより，従来の評価が最良の結果であることが明らかになるとともに，素因数分解したい数 n の構成と計算量の関係を考察することができる．

第 2 章では Shor の素因数分解アルゴリズムとその計算量の評価の方法を説明する．第 3 章では従来の計算量の評価を紹介する．第 4 章で本研究での精密な評価に必要な整数論の結果を説明し，第 5 章で実際に精密な評価を行う．最後に第 6 章で従来の評価と本研究の結果の比較をする．

2 Shor のアルゴリズム

Shor の素因数分解のアルゴリズムとその計算量の評価の方法を紹介する．素因数分解したい数を n とする．ここでは簡単のために n は二つの素数の積で $n = pq$ と表されている場合を考える．そうすると素因数分解のアルゴリズムは以下のようにまとめることができる．

1° : $\{1, 2, \ldots, n\}$ からランダムに一つ選び，その数を a とする．
2° : $\gcd(a, n) = 1$ ならば 3° へ行く．$\gcd(a, n) \neq 1$ ならば 1° へ戻る．
3° : a の mod$n$ に関する位数 r を求める．(量子コンピュータによる)
4° : 得られた位数 r が偶数ならば 5° へ行く．奇数ならば 1° へ戻る．
5° : $p' = \gcd(a^{r/2} + 1, n)$ と $q' = \gcd(a^{r/2} - 1, n)$ を求める．
6° : p', q' のいずれかが n ならば 1° へ戻る．そうでなければそれらが求める因数 p, q である．

次に，アルゴリズムの計算量の評価の方法を説明する [3]．アルゴリズムを 1 回実行して素因数分解に成功する確率を P_S とすると，十分大きい確率で正しく素因数分解するために必要なアルゴリズムの実行回数 N は

$$\forall \varepsilon > 0, \quad N \geq \log(1/\varepsilon)/P_S \tag{2.1}$$

を満たせばよい．ここで，確率 P_S を評価するために次のような事象を考える．

A_a : $\gcd(a, n) = 1$ となるような n 未満の数 a が得られる事象
A_r : 量子コンピュータによって正しい位数 r が得られる事象
A_e : 量子コンピュータによって得られた位数 r が偶数である事象
A_f : 得られた位数から正しい因数 p, q が得られる事象

これらの事象を用いると，確率 P_S は

$$\begin{aligned} P_S &= P(A_a \cap A_r) P(A_e \cap A_f \mid A_a \cap A_r) + P(A_a \cap A_r) P(A_e \cap A_f \mid \overline{A_a \cap A_r}) \\ &\geq P(A_a \cap A_r) P(A_e \cap A_f \mid A_a \cap A_r) \\ &= P(A_a) P(A_r) P(A_e \cap A_f \mid A_a \cap A_r) \end{aligned} \tag{2.2}$$

となる．ここで，確率 $P(A_a), P(A_r), P(A_e \cap A_f \mid A_a \cap A_r)$ を求めることで，式 (2.1) 及び式 (2.2) から必要なアルゴリズムの実行回数が得られる．

3 従来の計算量の評価

この章では，確率 $P(A_a), P(A_r), P(A_e \cap A_f \mid A_a \cap A_r)$ の従来の評価の方法を紹介する [3]．まず，確率 $P(A_a)$ は，$\gcd(a, n) = 1$ となるような n 未満の数 a が得られる確率であるから，Euler の関数を用いて

$$P(A_a) = \frac{\varphi(n)}{n-1}$$

と表すことができる．Euler の関数については

$$\liminf_{n \to \infty} \frac{\varphi(n) \log\log n}{n} = e^{-\gamma}$$

が成り立つことが知られている．ここで γ は Euler の定数である．したがって，十分大きい n に対して

$$P(A_a) = \frac{\varphi(n)}{n-1} \geq \frac{e^{-\gamma}}{\log\log n} \geq \frac{e^{-\gamma}}{\log n} = \frac{e^{-\gamma}\log_2 e}{\log_2 n} = \frac{\alpha}{\log_2 n} \tag{3.1}$$

が成り立つ．ここで α は n に依存しない定数である．

次に確率 $P(A_r)$ を求める．これは量子コンピュータによって正しい位数 r が得られる確率である．この確率は，r 未満で r と互いに素な数が得られる確率を用いて表すことができる．すなわち，確率 $P(A_a)$ の場合と同様にして

$$P(A_r) \geq \frac{\beta}{\log_2 n} \tag{3.2}$$

と表すことができる．ここで β は n に依存しない定数である．

最後に確率 $P(A_e \cap A_f \mid A_a \cap A_r)$ について説明する．一般に，k 種類の素数の積で $n = {p_1}^{e_1}{p_2}^{e_2}\ldots{p_k}^{e_k}$ と表される n に対して，量子コンピュータによって得られた位数 r が偶数であり，かつ得られた位数から正しい素因数が得られる確率 $P(A_e \cap A_f \mid A_a \cap A_r)$ は

$$P(A_e \cap A_f \mid A_a \cap A_r) \geq 1 - \frac{1}{2^{k-1}} \tag{3.3}$$

であることが知られている [4]．この確率を第 5 章でより精密に記述する．

以上のことから，アルゴリズムの計算量を評価することができる．式 (3.1), (3.2), (3.3) を式 (2.2) に代入すると，アルゴリズムを 1 回実行して素因数分解に成功する確率 P_S は

$$P_S \geq \left(1 - \frac{1}{2^{k-1}}\right)\frac{\alpha\beta}{(\log_2 n)^2} \tag{3.4}$$

となり，式 (2.1) より，十分大きい確率で正しく素因数分解するために必要なアルゴリズムの実行回数 N は

$$\forall\varepsilon > 0, \quad N \geq \frac{\log(1/\varepsilon)}{\alpha\beta(1-1/2^{k-1})}(\log_2 n)^2 \tag{3.5}$$

となる．すなわち，アルゴリズムの実行回数は，素因数分解したい数 n を 2 進数表示したときの桁数 $\log_2 n$ のオーダーになることがわかる．また，位数 r を求める量子コンピュータを構成するのに必要なゲートの数も $O(\log_2 n)$ であることが知られており，総合して Shor のアルゴリズムの計算量は $O(\log_2 n)$ である．

4 Shor のアルゴリズムに関連する整数論の結果

この章では，式 (3.3) をより精密に評価するために必要な整数論の結果を用意する．素数 p に対して，体 $\boldsymbol{Z}/p\boldsymbol{Z}$ の invertible element 全体を $(\boldsymbol{Z}/p\boldsymbol{Z})^\times$ とすると，$(\boldsymbol{Z}/p\boldsymbol{Z})^\times = \{1,2,\ldots,p-1\}$ は $p-1$ 個の元をもつ．このとき，よく知られた結果として

$$\left|\{a \in (\boldsymbol{Z}/p\boldsymbol{Z})^\times; r_p = d\}\right| = \varphi(d)$$

が成り立つ [5]．但し，$d \mid p-1$，r_p は a の mod$p$ に関する位数，$\varphi(\cdot)$ は Euler の関数である．

補題 4.1　素数 p に対して，$p-1 = 2^\tau\sigma$ $(\tau \geq 1, \sigma : odd)$ と書くことができ，

$$\left|\{a \in (\boldsymbol{Z}/p\boldsymbol{Z})^\times; r_p : odd\}\right| = \sigma \tag{4.1}$$

$$\left|\{a \in (\boldsymbol{Z}/p\boldsymbol{Z})^\times; r_p = 2^t s\,(s : odd)\}\right| = 2^{t-1}\sigma \tag{4.2}$$

が成り立つ．ここで，t は $1 \leq t \leq \tau$ の固定された数である．

(証明)　まず，$r_p = 2^t s\,(t \geq 0, s : odd)$ と書くと

$$r_p : odd,\ r_p \mid p-1 \Longleftrightarrow r_p \mid \sigma$$

が成り立つ．$r_p \mid p-1 = 2^t s \mid 2^\tau \sigma$ より $t \leq \tau$, $s \mid \sigma$ であり，$r_p = 2^t s$ が奇数であることから $t = 0$．したがって，$r_p = s$ となり，$s \mid \sigma$ から $r_p \mid \sigma$ が得られる．逆に，$r_p \mid \sigma$ ならば，r_p は σ の約数であるから r_p は奇数である．また，$p-1 = 2^\tau \sigma$ より $r_p \mid \sigma \Longrightarrow r_p \mid p-1$ が成り立つ。このことに注意すると

$$\begin{aligned}\left|\{a \in (\boldsymbol{Z}/p\boldsymbol{Z})^\times ; r_p : odd\}\right| &= \sum_{r_p \mid p-1,\, r_p : odd} \varphi(r_p) \\ &= \sum_{r_p \mid \sigma} \varphi(r_p) \\ &= \sigma\end{aligned}$$

となり，式 (4.1) が得られる．次に，$r_p = 2^t s$ のとき

$$r_p \mid p-1 \Longleftrightarrow s \mid \sigma$$

が成り立つ．仮定 $1 \leq t \leq \tau$ より $2^t s \mid 2^\tau \sigma \Longrightarrow s \mid \sigma$ であり，逆に $s \mid \sigma \Longrightarrow 2^t s \mid 2^\tau \sigma$ が成り立つからである．したがって，固定された $t\,(1 \leq t \leq \tau)$ に対して

$$\begin{aligned}\left|\{a \in (\boldsymbol{Z}/p\boldsymbol{Z})^\times ; r_p = 2^t s\,(s : odd)\}\right| &= \sum_{r_p \mid p-1,\, r_p = 2^t s} \varphi(r_p) \\ &= \sum_{s \mid \sigma} \varphi(2^t s) \\ &= \sum_{s \mid \sigma} \varphi(2^t)\varphi(s) \\ &= \varphi(2^t) \sum_{s \mid \sigma} \varphi(s) \\ &= 2^t \left(1 - \frac{1}{2}\right) \sigma \\ &= 2^{t-1}\sigma\end{aligned}$$

となり，式 (4.2) が得られる．

補題 4.2　$n = {p_1}^{e_1} \ldots {p_k}^{e_k}$ (p_i は素数，$i = 1, 2, \ldots, k$) に対して，$p_i - 1 = 2^{\tau_i}\sigma_i$ $(\tau_i \geq 1, \sigma_i : odd)$ と書くことができ，a の $\bmod n$ に関する位数を r，$\bmod p_i$ に関する位数を $r_{p_i} = 2^{t_{p_i}} s_{p_i}$ とすると

$$\left|\{a \in (\boldsymbol{Z}/n\boldsymbol{Z})^\times ; r : odd\}\right| = \prod_{i=1}^{k} \sigma_{p_i} \tag{4.3}$$

$$\left|\{a \in (\boldsymbol{Z}/n\boldsymbol{Z})^\times ; t_{p_1} = \cdots = t_{p_k} = l\}\right| = 2^{k(l-1)} \prod_{i=1}^{k} \sigma_{p_i} \tag{4.4}$$

が成り立つ．但し，$1 \leq l \leq \min(\tau_{p_1}, \ldots, \tau_{p_k})$ である．

(証明) Chinese Remainder Theorem より，$(\boldsymbol{Z}/n\boldsymbol{Z})^{\times} \cong (\boldsymbol{Z}/p_1\boldsymbol{Z})^{\times} \oplus \cdots \oplus (\boldsymbol{Z}/p_k\boldsymbol{Z})^{\times}$ が成り立つので，$a \in (\boldsymbol{Z}/n\boldsymbol{Z})^{\times}$ に対して

$$
\begin{aligned}
&r = \mathrm{lcm}\{r_{p_1}, \ldots, r_{p_k}\} \\
&r : odd \Longleftrightarrow r_{p_1}, \ldots, r_{p_k} : odd \\
&\left|(\boldsymbol{Z}/n\boldsymbol{Z})^{\times}\right| = \left|(\boldsymbol{Z}/p_1\boldsymbol{Z})^{\times}\right| \cdots \left|(\boldsymbol{Z}/p_k\boldsymbol{Z})^{\times}\right|
\end{aligned}
$$

が成り立つことに注意すると，式 (4.1) を用いて

$$
\begin{aligned}
&\left|\{a \in (\boldsymbol{Z}/n\boldsymbol{Z})^{\times}; r : odd\}\right| \\
&= \left|\{a \in (\boldsymbol{Z}/n\boldsymbol{Z})^{\times}; r_{p_1}, \ldots, r_{p_k} : odd\}\right| \\
&= \left|\{a \in (\boldsymbol{Z}/p_1\boldsymbol{Z})^{\times}; r_{p_1} : odd\}\right| \cdots \left|\{a \in (\boldsymbol{Z}/p_k\boldsymbol{Z})^{\times}; r_{p_k} : odd\}\right| \\
&= \prod_{i=1}^{k} \sigma_{p_i}
\end{aligned}
$$

となり，式 (4.3) が得られる．また，式 (4.2) を用いて

$$
\begin{aligned}
&\left|\{a \in (\boldsymbol{Z}/n\boldsymbol{Z})^{\times}; t_{p_1} = \cdots = t_{p_k} = l\}\right| \\
&= \left|\{a \in (\boldsymbol{Z}/p_1\boldsymbol{Z})^{\times}; t_{p_k} = l\}\right| \cdots \left|\{a \in (\boldsymbol{Z}/p_k\boldsymbol{Z})^{\times}; t_{p_k} = l\}\right| \\
&= 2^{k(l-1)} \prod_{i=1}^{k} \sigma_{p_i}
\end{aligned}
$$

となり，式 (4.4) が得られる．

定理 4.3 $\tau' = \min(\tau_{p_1}, \ldots, \tau_{p_k})$, $\tilde{\tau} = \sum_{i=1}^{k} \tau_{p_i}$ とおくと

$$
\left|\{a \in (\boldsymbol{Z}/n\boldsymbol{Z})^{\times}; t_{p_1} = \cdots = t_{p_k}\}\right| = \frac{2^k - 2 + 2^{k\tau'}}{2^k - 1} \prod_{i=1}^{k} \sigma_{p_i} \tag{4.5}
$$

であり

$$
\frac{\left|\{a \in (\boldsymbol{Z}/n\boldsymbol{Z})^{\times}; t_{p_1} = \cdots = t_{p_k}\}\right|}{\left|(\boldsymbol{Z}/n\boldsymbol{Z})^{\times}\right|} = \frac{1}{2^k - 1} \frac{2^k - 2 + 2^{k\tau'}}{2^{\tilde{\tau}}} \tag{4.6}
$$

が成り立つ．

(証明) 式 (4.5) の左辺を変形し，式 (4.3) および (4.4) を用いると

$$
\begin{aligned}
&\left|\{a \in (\boldsymbol{Z}/n\boldsymbol{Z})^{\times}; t_{p_1} = \cdots = t_{p_k}\}\right| \\
&= \left|\bigcup_{l=0}^{\tau'} \{a \in (\boldsymbol{Z}/n\boldsymbol{Z})^{\times}; t_{p_1} = \cdots = t_{p_k} = l\}\right| \\
&= \sum_{l=0}^{\tau'} \left|\{a \in (\boldsymbol{Z}/n\boldsymbol{Z})^{\times}; t_{p_1} = \cdots = t_{p_k} = l\}\right| \\
&= \left|\{a \in (\boldsymbol{Z}/n\boldsymbol{Z})^{\times}; t_{p_1} = \cdots = t_{p_k} = 0\}\right| + \sum_{l=1}^{\tau'} \left|\{a \in (\boldsymbol{Z}/n\boldsymbol{Z})^{\times}; t_{p_1} = \cdots = t_{p_k} = l\}\right| \\
&= \prod_{i=1}^{k} \sigma_{p_i} + \sum_{l=1}^{\tau'} \left(2^{k(l-1)} \prod_{i=1}^{k} \sigma_{p_i}\right)
\end{aligned}
$$

$$= \frac{2^k - 2 + 2^{k\tau'}}{2^k - 1} \prod_{i=1}^{k} \sigma_{p_i}$$

となり，右辺が得られる．また，

$$\left|(\boldsymbol{Z}/n\boldsymbol{Z})^{\times}\right| = \left|(\boldsymbol{Z}/p_1\boldsymbol{Z})^{\times}\right| \cdots \left|(\boldsymbol{Z}/p_k\boldsymbol{Z})^{\times}\right| = 2^{\tilde{\tau}} \prod_{i=1}^{k} \sigma_{p_i}$$

であることに注意すれば，式 (4.5) より式 (4.6) は明らか．

5 精密な計算量の評価

第 3 章の式 (3.3) で表される確率は，n の素因数の個数のみで表現されている．この章では，第 4 章での整数論の結果を用いて，素因数の個数だけでなく，素因数から定まる数を使って確率を精密に評価する．

補題 5.1 $n = {p_1}^{e_1} \ldots {p_k}^{e_k}$ (p_i は素数，$i = 1, 2, \ldots, k$) に対して，$p_i - 1 = 2^{\tau_i}\sigma_i$ ($\tau_i \geq 1$, σ_i は奇数)，$\tau' = \min(\tau_1, \ldots, \tau_k)$，$\tilde{\tau} = \sum_{i=1}^{k} \tau_i$ とすると

$$P(A_e \cap A_f \mid A_a \cap A_r) = 1 - \frac{1}{2^k - 1} \frac{2^k - 2 + 2^{k\tau'}}{2^{\tilde{\tau}}} \tag{5.1}$$

が成り立つ．

(証明) アルゴリズムのステップ 5° と 6° と位数の性質に注意すると

$$\begin{aligned} P(A_e \cap A_f \mid A_a \cap A_r) &= P\left(\{r : even\} \cap \{a^{r/2} \neq \pm 1 \pmod n\} \middle| A_a \cap A_r\right) \\ &= P\left(\{r : even\} \cap \{a^{r/2} \neq -1 \pmod n\} \middle| A_a \cap A_r\right) \\ &= 1 - P\left(\{r : odd\} \cup \{a^{r/2} = -1 \pmod n\} \middle| A_a \cap A_r\right) \end{aligned}$$

となる．ここで a の $\bmod n$ に関する位数を $r = 2^t s$，$\bmod p_i$ に関する位数を $r_i = 2^{t_i} s_i$ とする．ただし，$i = 1, 2, \ldots, k$，$t, t_i \geq 1$，s, s_i は奇数とする．すると確率 $P(A_e \cap A_f \mid A_a \cap A_r)$ はさらに変形でき

$$\begin{aligned} &P(A_e \cap A_f \mid A_a \cap A_r) \\ &= 1 - P\left(\{r : odd\} \cup \left(\bigcap_{i=1}^{k} \{a^{r/2} = -1 \pmod{p_i}\}\right) \middle| A_a \cap A_r\right) \\ &= 1 - P\left(\{t_1 = \cdots = t_k = 0\} \cup \left(\bigcap_{i=1}^{k} \{t_i = t\}\right) \middle| A_a \cap A_r\right) \\ &= 1 - P(t_1 = \cdots = t_k \mid A_a \cap A_r) \end{aligned}$$

となる．ここで式 (4.6) を用いると式 (5.1) が得られる．

定理 5.2 $n = {p_1}^{e_1} \ldots {p_k}^{e_k}$ (p_i は素数，$i = 1, 2, \ldots, k$) に対して，$p_i - 1 = 2^{\tau_i}\sigma_i$ ($\tau_i \geq 1$, σ_i は奇数)，$\tau' = \min(\tau_1, \ldots, \tau_k)$，$\tilde{\tau} = \sum_{i=1}^{k} \tau_i$ とすると，アルゴリズムを 1 回実行して素因数分解に成功する確率 P_S は

$$P_S \geq \left(1 - \frac{1}{2^k - 1} \frac{2^k - 2 + 2^{k\tau'}}{2^{\tilde{\tau}}}\right) \frac{\alpha\beta}{(\log_2 n)^2} \tag{5.2}$$

であり，十分大きい確率で正しく素因数分解するために必要なアルゴリズムの実行回数 N は

$$\forall \varepsilon > 0, \quad N \geq \frac{\log(1/\varepsilon)}{\alpha\beta\left(1 - \frac{1}{2^{k-1}}\frac{2^k - 2 + 2^{k\tau'}}{2^{\tilde{\tau}}}\right)} (\log_2 n)^2 \tag{5.3}$$

である．ここで α, β は n に依存しない定数である．

(証明)　式 (5.1) を式 (2.1) および (2.2) に用いる．

6　結果の比較

本論文で精密に評価した確率は，量子コンピュータによって得られた位数 r が偶数であり，かつ得られた位数から正しい素因数が得られる確率 $P(A_e \cap A_f \mid A_a \cap A_r)$ である．従来の評価では，$n = {p_1}^{e_1}{p_2}^{e_2}\ldots{p_k}^{e_k}$ とすると

$$P(A_e \cap A_f \mid A_a \cap A_r) \geq 1 - \frac{1}{2^{k-1}}$$

とされていたものに対して，$p_i - 1 = 2^{\tau_i}\sigma_i$ $(i = 1, 2, \ldots, k,\ \tau_i \geq 1,\ \sigma_i$ は奇数$)$，$\tau' = \min(\tau_1, \ldots, \tau_k)$，$\tilde{\tau} = \sum_{i=1}^{k} \tau_i$ とすることで

$$P(A_e \cap A_f \mid A_a \cap A_r) = 1 - \frac{1}{2^k - 1}\frac{2^k - 2 + 2^{k\tau'}}{2^{\tilde{\tau}}}$$

と精密に表すことができた．これらの式の間には次のような関係がある．

定理 6.1　$n = {p_1}^{e_1}\ldots{p_k}^{e_k}$ (p_i は素数，$i = 1, 2, \ldots, k$) に対して，$p_i - 1 = 2^{\tau_i}\sigma_i$ $(\tau_i \geq 1,\ \sigma_i$ は奇数$)$，$\tau' = \min(\tau_1, \ldots, \tau_k)$，$\tilde{\tau} = \sum_{i=1}^{k} \tau_i$ とすると

$$P(A_e \cap A_f \mid A_a \cap A_r) = 1 - \frac{1}{2^k - 1}\frac{2^k - 2 + 2^{k\tau'}}{2^{\tilde{\tau}}} \geq 1 - \frac{1}{2^{k-1}} \tag{6.1}$$

が成り立つ．等号成立は $\tau_1 = \cdots = \tau_k = 1$ のとき．

(証明)　式 (6.1) の不等式について

$$\begin{aligned}
&\frac{1}{2^{k-1}} - \frac{1}{2^k - 1}\frac{2^k - 2 - 2^{k\tau'}}{2^{\tilde{\tau}}} \\
&\geq \frac{1}{2^{k-1}} - \frac{1}{2^k - 1}\frac{2^k - 2 - 2^{k\tau'}}{2^{k\tau'}} \\
&= \left(\frac{1}{2^k} - \frac{1}{2^{k\tau'}}\right)\left(1 - \frac{1}{2^k - 1}\right) \\
&\geq 0
\end{aligned}$$

が成り立つ．

この定理により，従来の評価が最良の結果であり，確率 $P(A_e \cap A_f \mid A_a \cap A_r)$ の下限を $1 - \frac{1}{2^{k-1}}$ より大きくはできないことがわかる．また，$\tau_1 = \cdots = \tau_k = 1$ のときに確率 $P(A_e \cap A_f \mid A_a \cap A_r)$ が最小になることから，n を構成する素数が全て $(2 \cdot$ 奇数 $+ 1)$ の形をしているときに最も計算量を要することになる．

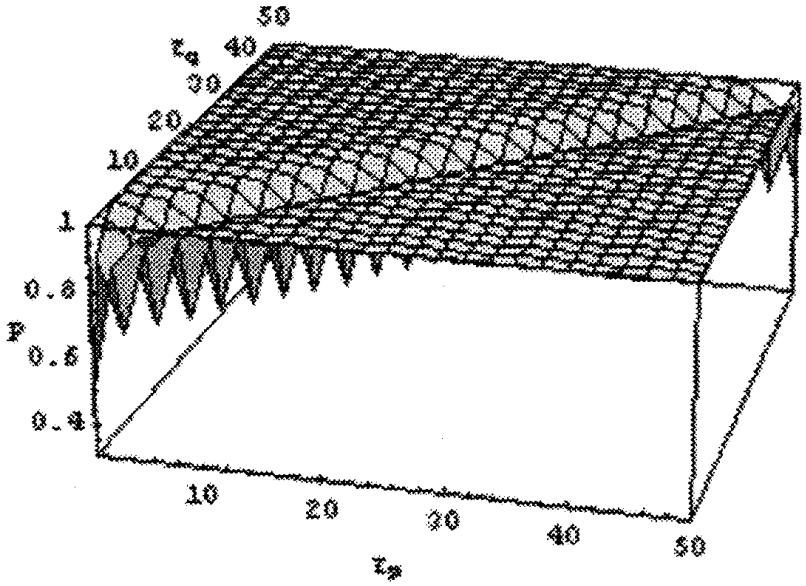

図1　τ_p および τ_q と確率 $P(A_e \cap A_f \mid A_a \cap A_r)$ の関係

n を構成する素数と確率 $P(A_e \cap A_f \mid A_a \cap A_r)$ の関係をグラフ（図 1）に示す．ここでは，n は 2 つの素数の積で $n = pq$ と表されている場合を考え，$p-1 = 2^{\tau_p}\sigma_p$, $q-1 = 2^{\tau_q}\sigma_q$ （$\tau_p, \tau_q \geq 1$，σ_p, σ_q は奇数）とする．このときの確率は，従来の評価では式 (3.3) より

$$P(A_e \cap A_f \mid A_a \cap A_r) \geq \frac{1}{2}$$

である．一方，本研究の精密な評価では，式 (5.1) より

$$P(A_e \cap A_f \mid A_a \cap A_r) = 1 - \frac{1}{3}\frac{2 + 2^{2\min(\tau_p,\tau_q)}}{2^{\tau_p+\tau_q}}$$

となる．図からわかるように，$\tau_p = \tau_q = 1$ のとき確率 1/2 で最小となっている．また，一般に $\tau_p = \tau_q$ のときに確率 $P(A_e \cap A_f \mid A_a \cap A_r)$ は比較的小さくなり，$\tau_p \neq \tau_q$ で急激に確率 1 に近づくことがわかる．

また，n が 2 つの素数の積で表される場合に限り，式 (3.1) の $\gcd(a, n) = 1$ となるような n 未満の数 a が得られる確率は簡単にすることができる．

補題 6.2　$n = pq$ （p, q は素数）とする．十分大きい n に対して，$\gcd(a, n) = 1$ となるような n 未満の数 a が得られる確率 $P(A_a)$ は

$$\forall \varepsilon > 0, \quad P(A_a) = \frac{\varphi(n)}{n} \geq \frac{1}{2} - \varepsilon$$

で与えられる．

（証明）　$n = pq$ （p, q は素数）とすると，Euler の関数は $\varphi(n) = n\left(1 - \frac{1}{p}\right)\left(1 - \frac{1}{q}\right)$ であるから

$$\frac{\varphi(n)}{n} = \left(1 - \frac{1}{p}\right)\left(1 - \frac{1}{q}\right) \geq \left(1 - \frac{1}{2}\right)\left(1 - \frac{1}{q}\right) = \frac{1}{2}\left(1 - \frac{1}{q}\right)$$

となる．q についても同様にすると

$$\frac{\varphi(n)}{n} \geq \frac{1}{2}\left(1-\frac{1}{p}\right),\ \frac{1}{2}\left(1-\frac{1}{q}\right)$$

が得られる．ここで，$n \to \infty \Longleftrightarrow (p \to \infty$ または $q \to \infty)$ に注意すると，任意の ε に対して，n を十分大きくすると

$$\frac{\varphi(n)}{n} > \frac{1}{2} - \varepsilon$$

となる．

系 6.3　$n = pq$ (p, q は素数) とする．$p-1 = 2^{\tau_p}\sigma_p$, $q-1 = 2^{\tau_q}\sigma_q$, $\tau' = \min(\tau_p, \tau_q)$ (但し，$\tau_p, \tau_q \geq 1$, σ_p, σ_q は奇数) とすると，アルゴリズムを 1 回実行して素因数分解に成功する確率 P_S は

$$P_S \geq \frac{\alpha}{2\log_2 n}\left(1 - \frac{1}{3}\frac{2+2^{2\tau'}}{2^{\tau_p+\tau_q}}\right)$$

であり，十分大きい確率で確率で正しく素因数分解するために必要なアルゴリズムの実行回数 N は

$$\forall \varepsilon > 0, \quad N \geq \frac{2\log(1/\varepsilon)}{\alpha\left(1 - \frac{1}{3}\frac{2+2^{2\tau'}}{2^{\tau_p+\tau_q}}\right)}\log_2 n$$

である．

参考文献

[1] P.W.Shor, Algorithms for quantum computation: Discrete log and factoring, Proc. of the 35th Annual IEEE Symp. on Foundations of Computer Science, pp.124-134, 1994.

[2] P.W.Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM Journal on Computing, vol.26, no.5, pp.1484-1509, 1997.

[3] 上坂吉則, 量子コンピュータの基礎数理, コロナ社, 2000.

[4] A.Ekert and R.Jozsa, Quantum computation and Shor's factoring algorithm, Rev.Mod.Phys., 68, 3, pp.733-753, 1996.

[5] G.H.Hardy and E.M.Wright, An introduction to the Theory of Numbers, Fifth Edition, Oxford Science Publications, 1979.